

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software).

<<http://www.ati.es/novatica/>>
<<http://www.ati.es/reicis/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AI2**, **ASTIC**, **RITSI** e **HispanLinux**, junto a la que participa en **Prolnnova**.

Consejo Editorial

Ignacio Aguiló Sousa, Guillem Alsina González, María José Escalona Cuaresma, Rafael Fernández Calvo (presidente del Consejo), Jaime Fernández Martínez, Luis Fernández Sanz, Didac López Viñas, Celestino Martín Alonso, José Onofre Montes Andrés, Francesc Noguera Puig, Ignacio Pérez Martínez, Andrés Pérez Payeras, Viktu Pons i Colomer, Juan Carlos Vigo López

Coordinación Editorial

Llorenç Pagés Casas <lpages@ati.es>

Composición y autoedición

Jorge Llácer Gil de Ramales

Traducciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>>

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas - Coordinadores

Acceso y recuperación de la información

José María Gómez Hidalgo (Optenet), <jmgomez@yahoou.es>

Manuel J. María López (Universidad de Huelva), <manuel.mana@dieisia.uhu.es>

Administración Pública electrónica

Francisco López Crespo (MAE), <flc@ati.es>

Sebastià Justicia Pérez (Diputación de Barcelona), <sjusticia@ati.es>

Arquitecturas

Enrique F. Torres Moreno (Universidad de Zaragoza), <enrique.torres@unizar.es>

José Filich Cardo (Universidad Politécnica de Valencia), <jfilich@disca.upv.es>

Auditoría SITIC

Marina Tourino Trolitio, <marinatourino@marinatourino.com>

Manuel Palao García-Suñto (ATI), <manuel@palao.com>

Derecho y tecnologías

Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV), <isabel.hernando@ehu.es>

Elena Davara Fernández de Marcos (Davara & Davara), <edavara@davara.com>

Enseñanza Universitaria de la Informática

Cristóbal Pareja Flores (DSIP-UCM), <cpareja@sisip.ucm.es>

J. Ángel Velázquez Iturbide (DLSI, URJC), <angel.velazquez@urjc.es>

Entorno digital persona

Andrés Marín López (Univ. Carlos III), <amarin@it.uc3m.es>

Diego Gachet Páez (Universidad Europea de Madrid), <gachet@uem.es>

Estandares Web

Encarna Quesada Ruiz (Viratli), <encarna.quesada@viratli.com>

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería), <jcarco@gmail.com>

Gestión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young), <juan.baiget@ati.es>

Informática y Filosofía

José Ángel Olivás Valls (Escuela Superior de Informática, UCLM), <josangel.olivas@uclm.es>

Roberto Feltrero Oreja (UNED), <rfeltrero@gmail.com>

Informática Gráfica

Miguel Chover Selles (Universitat Jaume I de Castellón), <mchover@lsi.uji.es>

Roberto Vívio Hernández (Eurographics, sección española), <rvivio@dsic.upv.es>

Ingeniería del Software

Javier Dolado Cosín (DLSI-UPV), <dolado@si.ehu.es>

Daniel Rodríguez García (Universidad de Alcalá), <daniel.rodriguez@uah.es>

Inteligencia Artificial

Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV), <vbotti@inglada@dsic.upv.es>

Interacción Persona-Computador

Pedro M. Latore Andrés (Universidad de Zaragoza, AIP), <platore@unizar.es>

Francisco L. Gutiérrez Vela (Universidad de Granada, AIP), <fgutierrez@ugr.es>

Lengua e Informática

M. del Carmen Ugarte García (ATI), <cugarte@ati.es>

Lenguajes Informáticos

Oscar Belmonte Fernández (Univ. Jaime I de Castellón), <obelfern@lsi.uji.es>

Inmaculada Coma Tatay (Univ. de Valencia), <inmaculada.coma@uv.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo), <xgg@uvigo.es>

Manuel Palomar (Univ. de Alicante), <mpalomar@dsic.ua.es>

Mundo estudiantil y jóvenes profesionales

Federico G. Mon Trotti (RITSI), <gmon.fede@gmail.com>

Mikel Salazar Peña (Asociación Jóvenes Profesionales, Junta de ATI Madrid), <mikelboi_uni@yahoo.es>

Profesión Informática

Rafael Fernández Calvo (ATI), <rfcalvo@ati.es>

Miguel Sarrías Grilo (ATI), <miguels@sarries.net>

Redes y servicios telemáticos

José Luis Marco Lázaro (Univ. de Girona), <joseluis.marco@udg.es>

Juan Carlos López López (UCLM), <juancarlos.lopez@uclm.es>

Robótica

José Cortés Arenas (Sopra Group), <jccortes@gmail.com>

Juan González Gómez (Universidad CARLOS III), <juan@iearobotics.com>

Seguridad

Javier Arellano Bertollín (Univ. de Deusto), <jaarellito@deusto.es>

Javier López Muñoz (ETS Informática-UMA), <jlm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Fuente Altoro (DIT-UPM), <gaalonso@puente@dit.upm.es>

Software Libre

Jesús M. González Barahona (GSYC - URJC), <jgb@gsyc.es>

Isra Herráiz Tabernero (Universidad Politécnica de Madrid), <isra@herraz.org>

Tecnología de Objetos

Jesús García Moine (DIS-UM), <jmoine@um.es>

Gustavo Rossi (LFLIA-UNLP Argentina), <gustavo@sol.info.unlp.edu.ar>

Tecnologías para la Educación

Juan Manuel Dodero Beardo (UC3M), <dodero@inf.uc3m.es>

César Pablo Córcoles Briongo (UOC), <ccorcoles@uoc.edu>

Tecnologías y Empresa

Didac López Viñas (Universitat de Girona), <didac.lopez@ati.es>

Francisco Javier Cantais Sánchez (Indra Sistemas), <fjcantais@gmail.com>

Tendencias tecnológicas

Alonso Álvarez García (TID), <aaad@tid.es>

Gabriel Martí Fuentes (Interbits), <gabi@atinet.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga), <aguayo.guevara@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos.

Novática permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º dcha., 28006 Madrid

Tlfm. 914029391; fax 913093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Av. del Reino de Valencia 23, 46005 Valencia

Tlfm. 963740173 <novatica_valencia@ati.es>

Administración y Redacción ATI Cataluña

Via Laietana 46, ppal. 1º, 08003 Barcelona

Tlfm. 934125235; fax 934127713 <secregen@ati.es>

Redacción ATI Aragón

Lagasca 3, 5º B, 50006 Zaragoza

Tlfm. / fax 913238181 <secreara@ati.es>

Redacción ATI Andalucía

<secreand@ati.es>

Redacción ATI Galicia

<secregal@ati.es>

Subscripción y Ventas

<novatica.subscriptions@atinet.es>

Publicidad

Padilla 66, 3º dcha., 28006 Madrid

Tlfm. 914029391; fax 913093685 <novatica@ati.es>

Imprenta: Derra S.A., Juan de Austria 66, 08005 Barcelona

Depósito legal: B 15.154-1975 - ISSN: 0211-2124; CODEN NOVACE

Portada: Escalera de color - Concha Arias Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

Nº 221, enero-febrero 2013, año XXXIX

editorial

Ingeniería del Software en un momento de cambios y crisis > 02

en resumen

Ingeniería del Software y sistemas de engranaje múltiple > 02

Llorenç Pagés Casas

noticias de IFIP

Reunión del Board de IFIP > 03

Ramon Puigjaner Trepal

IFIP TC3 en el congreso WSIS+ 10 de UNESCO > 04

Carlos Delgado Kloos

Ramon López de Mántaras obtiene el Premio Nacional de Informática 2012 > 05

monografía

Técnicas avanzadas de desarrollo modular

Editores invitados: Mercedes Amor Pinilla, Lidia Fuentes Fernández, Mónica Pinto Alarcón

Presentación. Enfoques actuales para el desarrollo de software modular > 06

Mercedes Amor Pinilla, Lidia Fuentes Fernández, Mónica Pinto Alarcón

Análisis de la modularidad en sistemas software mediante un proceso de minería de aspectos > 09

José María Conejero Manzano, Juan Hernández Núñez

La Programación Orientada a Aspectos como mecanismo para aumentar la modularidad en la implementación de aplicaciones > 19

Francisco Ortín Soler

Enfoque basado en MDA para apoyar evoluciones seguras en sistemas orientados a aspectos > 25

Paulo F. Pires, Flávia C. Delicato, Jesús Martín Talavera Portocarrero

Modularidad en transformaciones de modelos > 34

Jesús J. García Molina, Jesús Sánchez Cuadrado

Separación avanzada de conceptos en el desarrollo de aplicaciones web > 42

Antonia Mª Reina Quintero, Rafael Corchuelo Gil, Miguel Toro Bonilla

Variabilidad en Ingeniería de Software Empotrado > 51

Salvador Trujillo González, David Benavides Cuevas

secciones técnicas

Enseñanza Universitaria de la Informática

Los conocimientos que deberán incluir los planes y programas de estudio de informática los próximos años en México (Modelo Curricular) > 54

Lourdes Sánchez-Guerrero, Rafaela Blanca Silva, José Raymundo Lira-Cortés

Mundo estudiantil y jóvenes profesionales

Nela: Aprende a escribir usando Braille > 57

Enrique Matías Sánchez, Inmaculada Plaza García, Nuria Tregón Martín

GeoTask: Servicios basados en localización para sistemas Android > 62

Francisco Javier Martín Otero

daf-collage: Un proyecto innovador en la enseñanza del idioma alemán > 66

Francisco Javier Rodríguez López, Simeón Ruiz Romero

Cormoran: Un framework de persistencia para Python > 69

Jaime Gil de Sagredo Luna

Referencias autorizadas > 71

Sociedad de la Información

Programar es crear

El problema del supermercado (Competencia UTN-FRC 2011, problema E, solución) > 77

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

asuntos interiores

Coordinación editorial / Programación de Novática / Socios Institucionales > 79

Tema del próximo número: "Lenguajes de programación"

Julio Javier Castillo, Diego Javier Serrano, Marina Elizabeth Cárdenas

Laboratorio de Investigación de Software MsLabs, Dpto. Ing. en Sistemas de Información, Facultad Regional Córdoba - Universidad Tecnológica Nacional (Argentina)

<jotacastillo@gmail.com>, <diegojserrano@gmail.com>, <ing.marinacardenas@gmail.com>

El problema del supermercado

El enunciado de este problema apareció en el número 218 de *Novática* (julio-agosto 2012, p.77)

El problema planteado consiste en determinar el instante de tiempo en que un supermercado atiende a su último cliente. Para ello, se tienen los datos de la cantidad de cajas que posee el supermercado, la cantidad de clientes, y el tiempo de atención de cada cliente que dependerá de la cantidad de artículos comprados.

Una manera de resolver este problema es plantearlo como un problema de simulación en la cual se diseñan las estructuras de datos que permitan modelar el problema y su dinámica de la manera más efectiva. Una lectura cuidadosa del enunciado nos permite determinar que se trata de un sistema determinista puesto que no interviene ninguna variable aleatoria, por lo cual la salida del sistema estará dada por una función deducible a partir de sus entradas.

En la siguiente solución se utilizan vectores para modelar las cajas del supermercado y se emplean colas para modelar los clientes en espera en cada caja. De esta manera, el supermercado es un vector de *colas* donde en cada

posición se encola un *cliente* que es representado por un número entero que indica el tiempo de atención que una cajera necesita para atender a ese *cliente*.

La solución hace uso solamente de dos clases, la clase Principal y la clase Caja, para realizar un balance óptimo entre la complejidad y claridad de la solución, y entre los principios de diseño de la Programación Orientada a Objetos y la eficiencia de la solución.

La variable utilizada C representa la cantidad de casos de prueba del problema, mientras que T representa la cantidad de segundos desde la apertura del supermercado hasta que el cliente elige una cola para pagar y A representa la cantidad de artículos comprados.

Nótese que el problema expresamente menciona que hay 10 cajas en el supermercado y ello podría generalizarse fácilmente a N-cajas. La solución propuesta trabaja con un vector de 10 cajas como lo prevé el enunciado pero este parámetro puede ser fácilmente

modificado solamente cambiando el valor de la variable entera J.

La clase Caja nos permite modelar una caja como una lista de enteros (horario de llegada de los clientes a la caja), y cuenta con métodos que nos permiten agregar clientes, y conocer cuál es la cola con menor cantidad de clientes (método `cantidadClientes()`), lo que nos permitirá determinar en qué caja se deberá encolar el próximo cliente.

La clase Principal es la que se encarga de la entrada de datos, y en base a ellos, de la creación de las cajas con sus respectivos clientes. A su vez, ubica cual es la caja con menor cantidad de clientes (primer ciclo `for`) y determina el horario de cierre del supermercado buscando el máximo tiempo de atención en cada una de las listas asociadas a las cajas (segundo ciclo `for`).

A continuación se expone el código de la solución del problema en el lenguaje de programación Java.

```
import java.util.ArrayList;
import java.util.Scanner;

class Caja {
    private final ArrayList<Integer> clientes = new ArrayList<>();
    private int maxSalida = 0;

    public void agregarCliente(int T, int A) {
        Integer horaSalida = 20 + A + (estaVacía(T) ? T : maximaSalida() + 1);
        clientes.add(horaSalida);
        maxSalida = horaSalida;
    }

    public int cantidadClientes(int hora) {
        int c = 0;
        for (Integer horaSalida : clientes)
            if (horaSalida >= hora)
                c++;
        return c;
    }

    public int maximaSalida() {
```

```
        return maxSalida;
    }

    public boolean estaVacia(int hora) {
        return cantidadClientes(hora) == 0;
    }
}

public class Principal {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int C = sc.nextInt();
        while (C > 0) {
            int N = sc.nextInt();
            int J = 10; //cantidad de cajas del supermercado

            Caja[] cajas = new Caja[J];
            for (int i = 0; i < J; i++) {
                cajas[i] = new Caja();
            }
            while (N > 0) {
                int T = sc.nextInt();
                int A = sc.nextInt();

                Caja menorCola = cajas[0];
                int cantidadClientes = menorCola.cantidadClientes(T);
                for (int i = 1; i < J; i++) {
                    int cantidad = cajas[i].cantidadClientes(T);
                    if (cantidad < cantidadClientes) {
                        cantidadClientes = cantidad;
                        menorCola = cajas[i];
                    }
                }
                menorCola.agregarCliente(T, A);
            }

            int ultimoCierre = cajas[0].maximaSalida();
            for (int i = 1; i < J; i++) {
                if (cajas[i].maximaSalida() > ultimoCierre)
                    ultimoCierre = cajas[i].maximaSalida();
            }
            System.out.println(ultimoCierre);
        }
    }
}
```